ÉCOLE SUPÉRIEURE D'ÉLECTRICITÉ

Simulation of multi-formalism models with ModHel'X

Cécile Hardebolle, Frédéric Boulanger

11 April 2008

Cécile HARDEBOLLE
Supélec – Computer Science Department
✉ cecile.hardebolle@supelec.fr

SYSTEM@TIC PARIS-REGION Pôle de compétitivité

Agenda

Introduction

- Context
  - Русский **Heterogeneous systems:** software/hardware, digital/analog, IPs…
  - 精忠 **Multiple modeling formalisms:** level of refinement, aspect, domain…

Introduction

- Context
  - Русский 精忠 Heterogeneous systems: software/hardware, digital/analog, IPs…
  - Multiple modeling formalisms: level of refinement, aspect, domain…

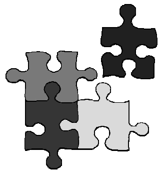
Global reasoning is impossible!

Introduction

- Context
  -

  - Heterogeneous systems: software/hardware, digital/analog, IPs…
  - Multiple modeling formalisms: level of refinement, aspect, domain…

Global reasoning is impossible!



Multi-formalism modeling =
allow the use of several modeling languages in a model

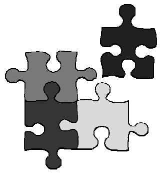
- Objective: having a global model of the designed system all along the design cycle
  - Design, test, verification, validation, …

Introduction

- Context
  - Heterogeneous systems: software/hardware, digital/analog, IPs…
  - Multiple modeling formalisms: level of refinement, aspect, domain…

Русский 精忠

Global reasoning is impossible!



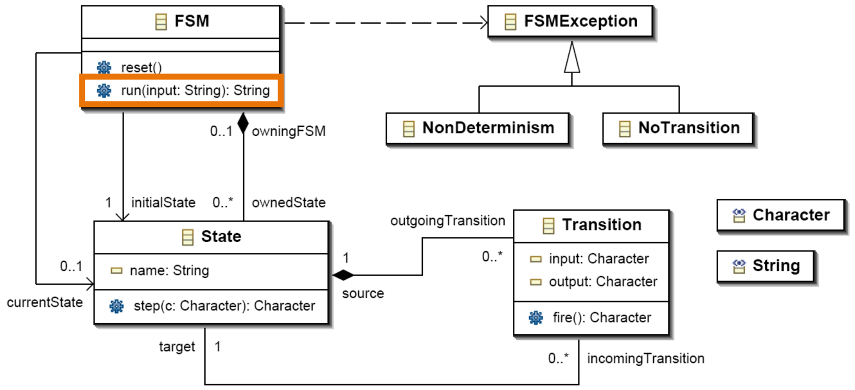
Multi-formalism modeling =
allow the use of several modeling languages in a model

- Objective: having a global model of the designed system
  all along the design cycle
  - Design, test, verification, validation, …
- Main issues
  1. Describe the semantics of a modeling language precisely (executable)
  2. Define the semantics of a combination of modeling languages in a model

Existing approaches & motivations

1. Defining the semantics of modeling languages

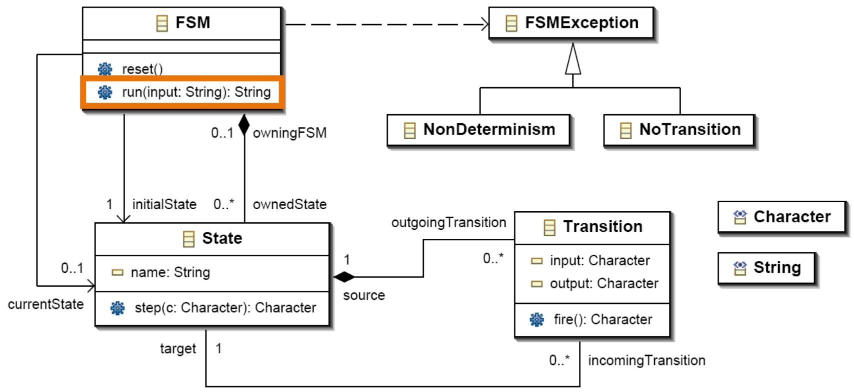
[Kermeta] ▸ Ad-hoc meta-model + execution operations (imperative semantics)



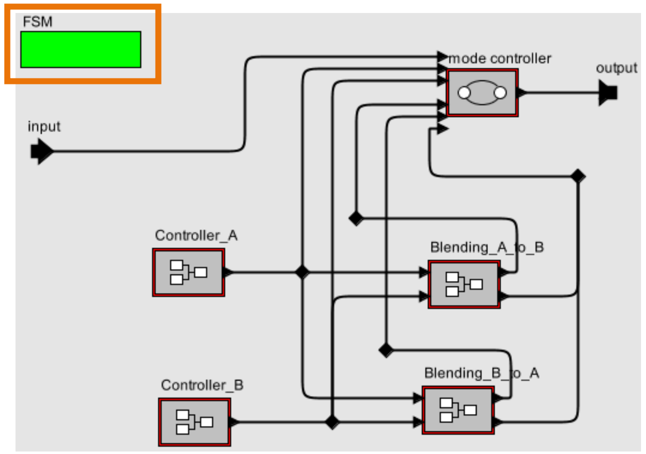
Existing approaches & motivations

1. Defining the semantics of modeling languages

[Kermeta] ▸ Ad-hoc meta-model + execution operations (imperative semantics)



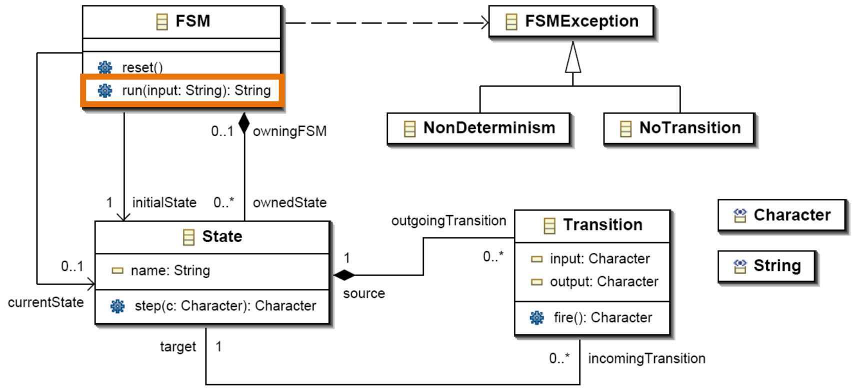
[PtolemyII] ▸ Fixed component-oriented abstract syntax + Model of Computation (MoC)



Existing approaches & motivations

1. Defining the semantics of modeling languages

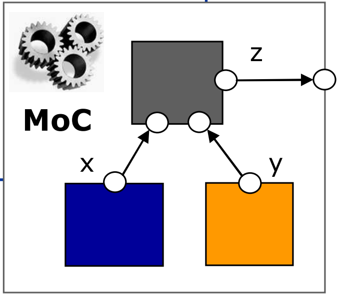
[Kermeta] ▸ Ad-hoc meta-model + execution operations (imperative semantics)



[PtolemyII] ▸ Fixed component-oriented abstract syntax
+ Model of Computation (MoC)

Set of rules that define the behavior of the model
by combining the behaviors of its components

= "way of interpreting the model"



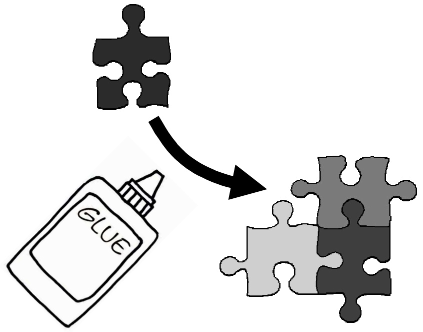
Existing approaches & motivations

1. Defining the semantics of modeling languages

[Kermeta] ▸ Ad-hoc meta-model + execution operations (imperative semantics)

[PtolemyII] ▸ Fixed component-oriented abstract syntax + Model of Computation (MoC)

How to “glue” heterogeneous parts together in a model?

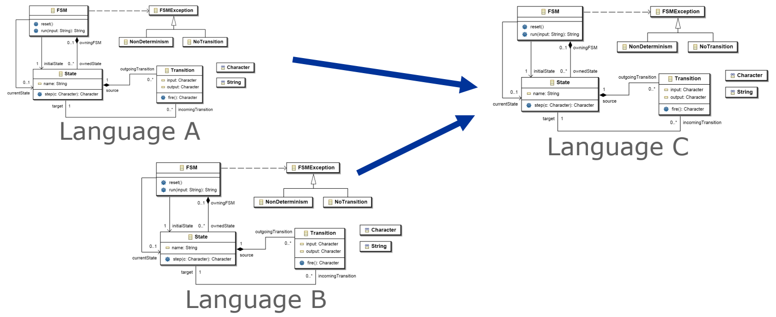


Existing approaches & motivations

2. Combining modeling languages in a model
   - Transformation toward a union meta-model

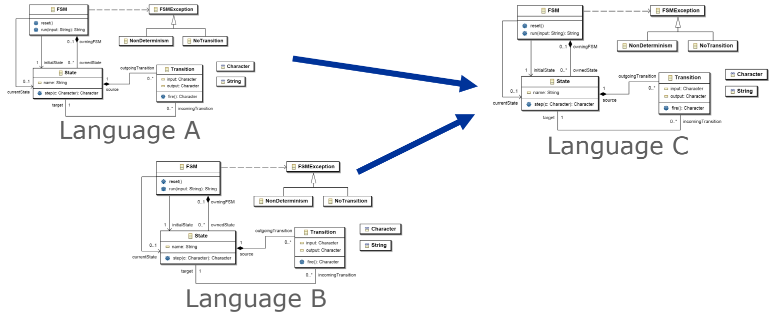
Existing approaches & motivations

2. Combining modeling languages in a model
   - Transformation toward a union meta-model
   - [ATOM^3] Transformation toward one of the modeling languages

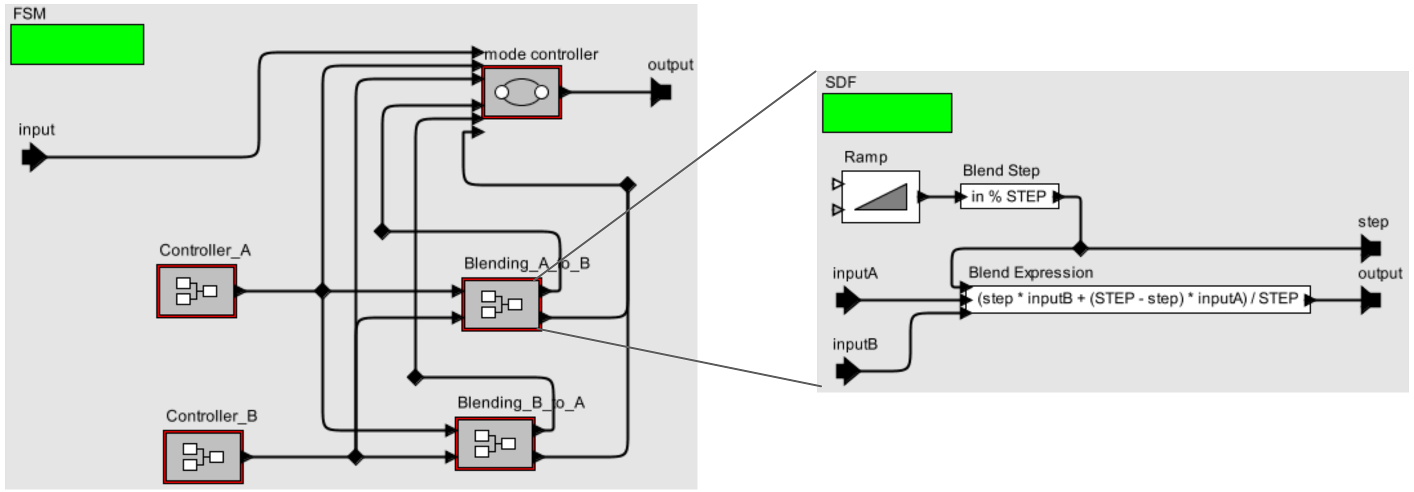


Existing approaches & motivations

2. Combining modeling languages in a model
   - Transformation toward a union meta-model
   - [ATOM^3] Transformation toward one of the modeling languages



   - [PtolemyII] Hierarchical layers using different Models of Computation (MoCs)

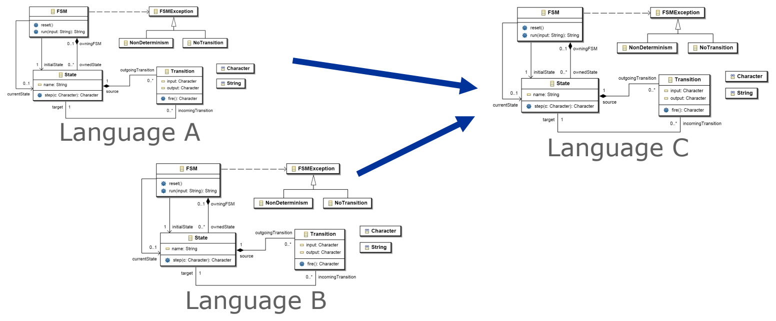


Existing approaches & motivations

2. Combining modeling languages in a model

- Transformation toward a union meta-model

[ATOM3] Transformation toward one of the modeling languages



[PtolemyII] Hierarchical layers using different Models of Computation (MoCs)

Issue: predefined & implicit glue between layers

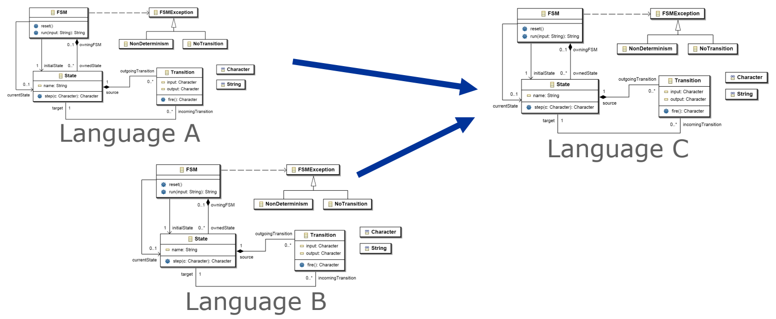
Modification of the models
in order to obtain an adapted glue

Existing approaches & motivations

2. Combining modeling languages in a model

- Transformation toward a union meta-model

[ATOM³] Transformation toward one of the modeling languages



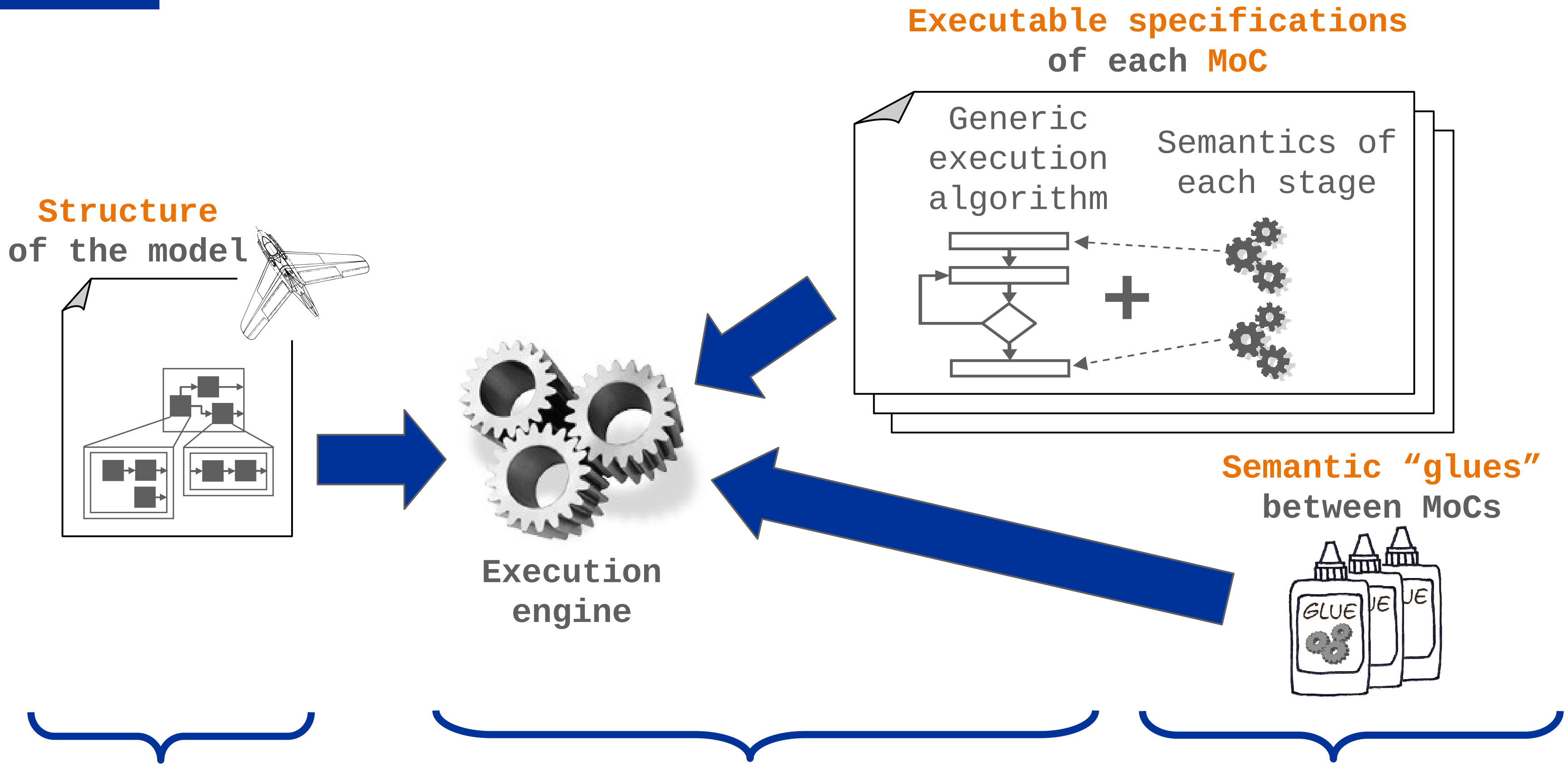
[PtolemyII] Hierarchical layers using different Models of Computation (MoCs)

Issue: predefined & implicit glue between layers

Modification of the models
in order to obtain an adapted glue

ModHel'X = hierarchical layers + MoCs
+ explicit specification of glues

General architecture of ModHel'X



Generic
component-oriented
& hierarchical
abstract syntax

(MOF meta-model)

Generic
model of execution

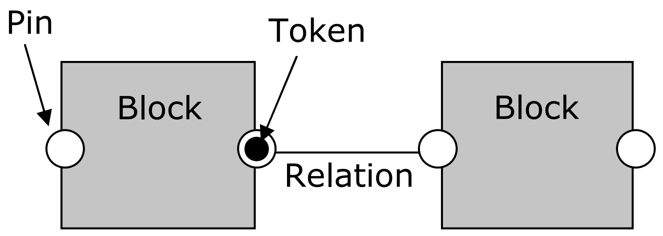
Language for
specifying semantics

Agenda

Agenda

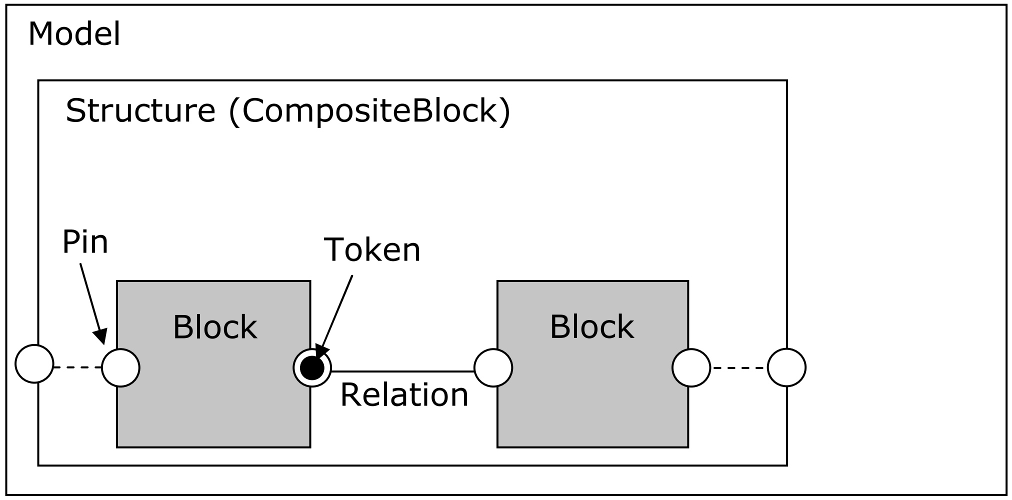
Generic block-oriented abstract syntax

- A few basic concepts: blocks, pins and relations



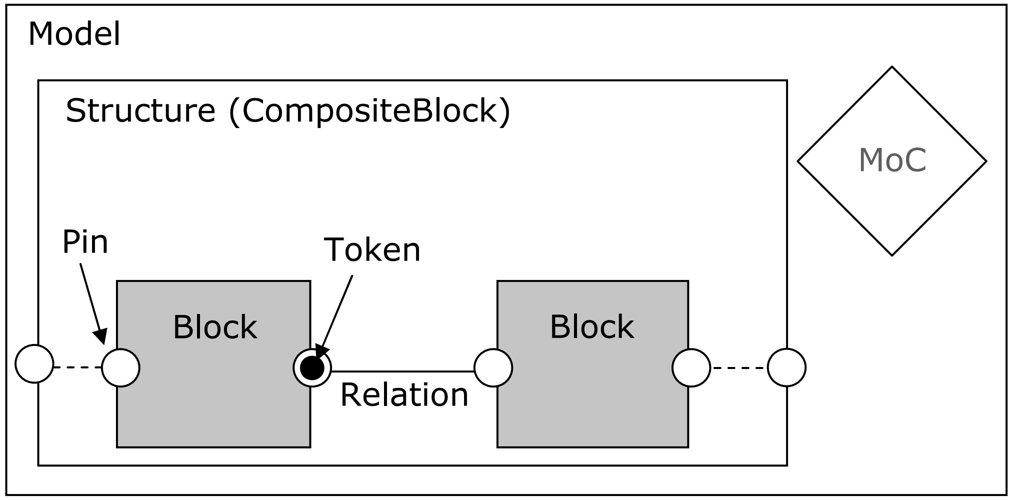
Generic block-oriented abstract syntax

- Structure of a model



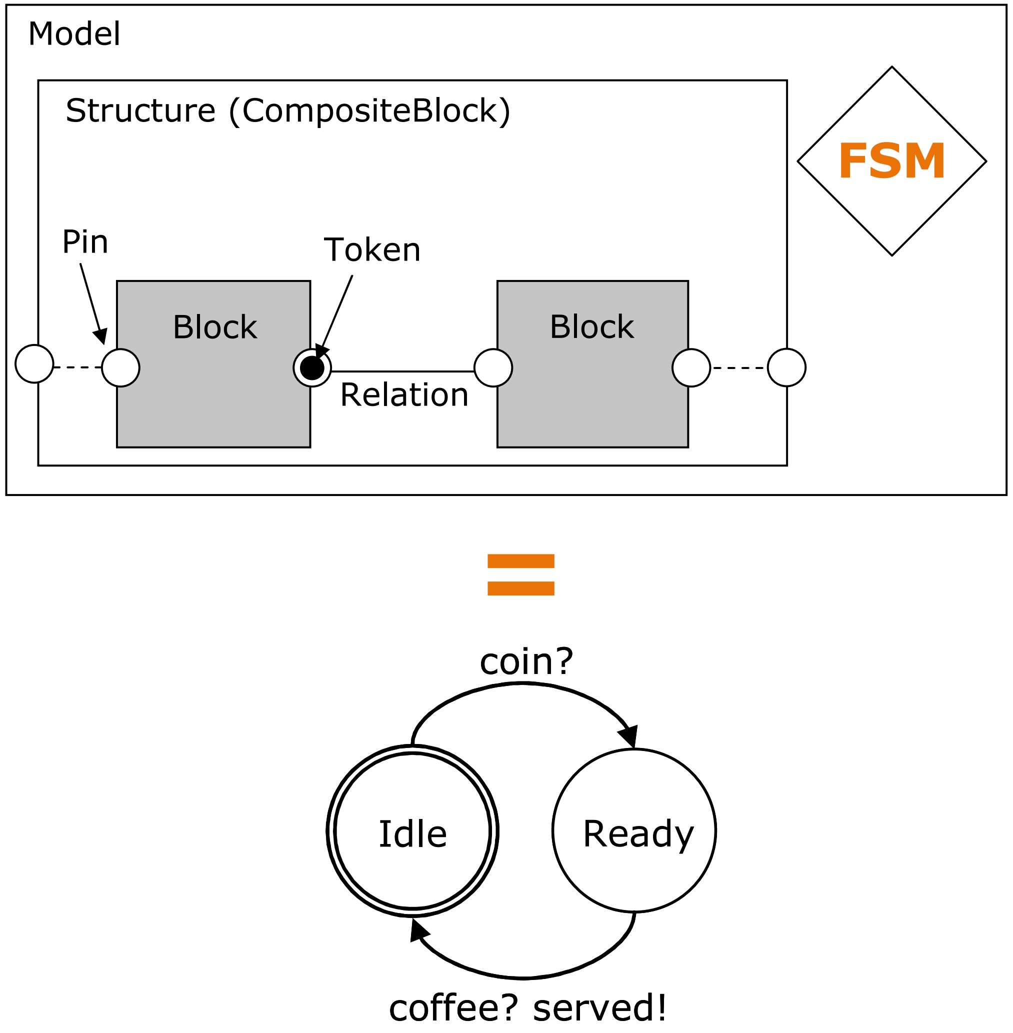
Model of Computation (MoC)

- Associated semantics: the MoC



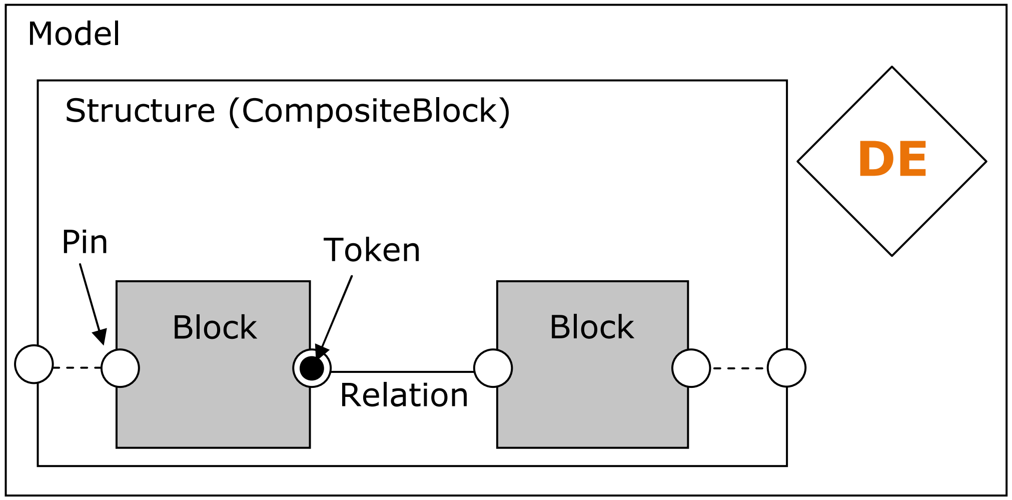
Model of Computation (MoC)

- A given structure, 2 different semantics with 2 different MoCs !

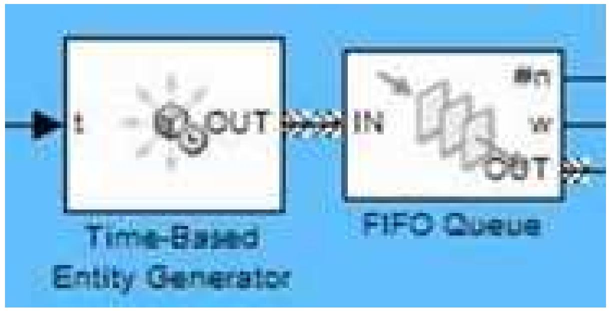


Model of Computation (MoC)

- A given structure, 2 different semantics with 2 different MoCs !



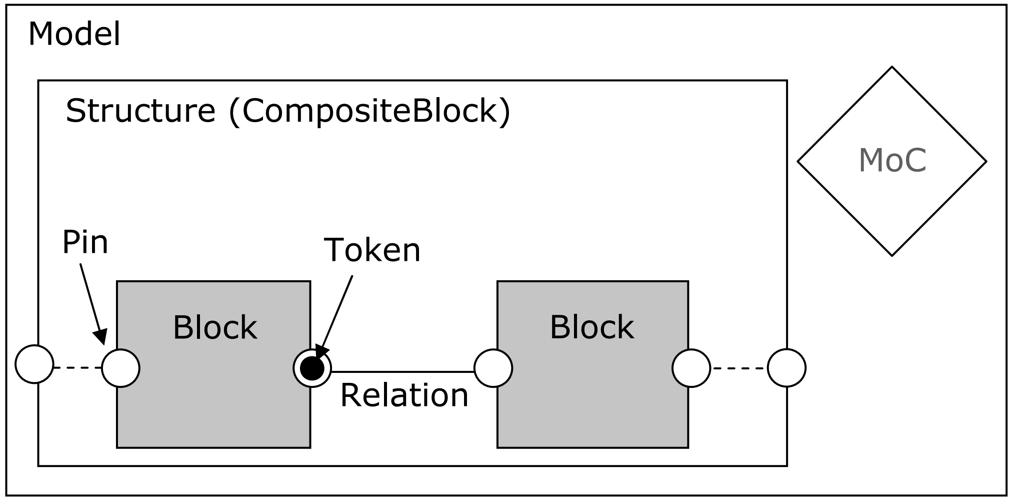
=



(TheMathworks SimEvents)

Model of Computation (MoC)

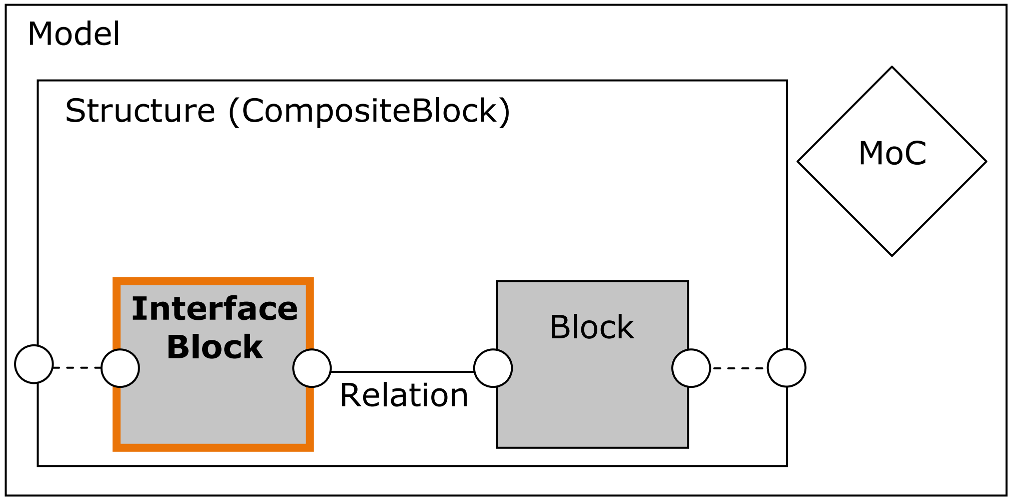
- A given structure, 2 different semantics with 2 different MoCs !



  - Specialization of the abstract syntax allowed to
    - Represent particular concepts used in certain MoCs
    - Constrain the structure of models for particular MoCs

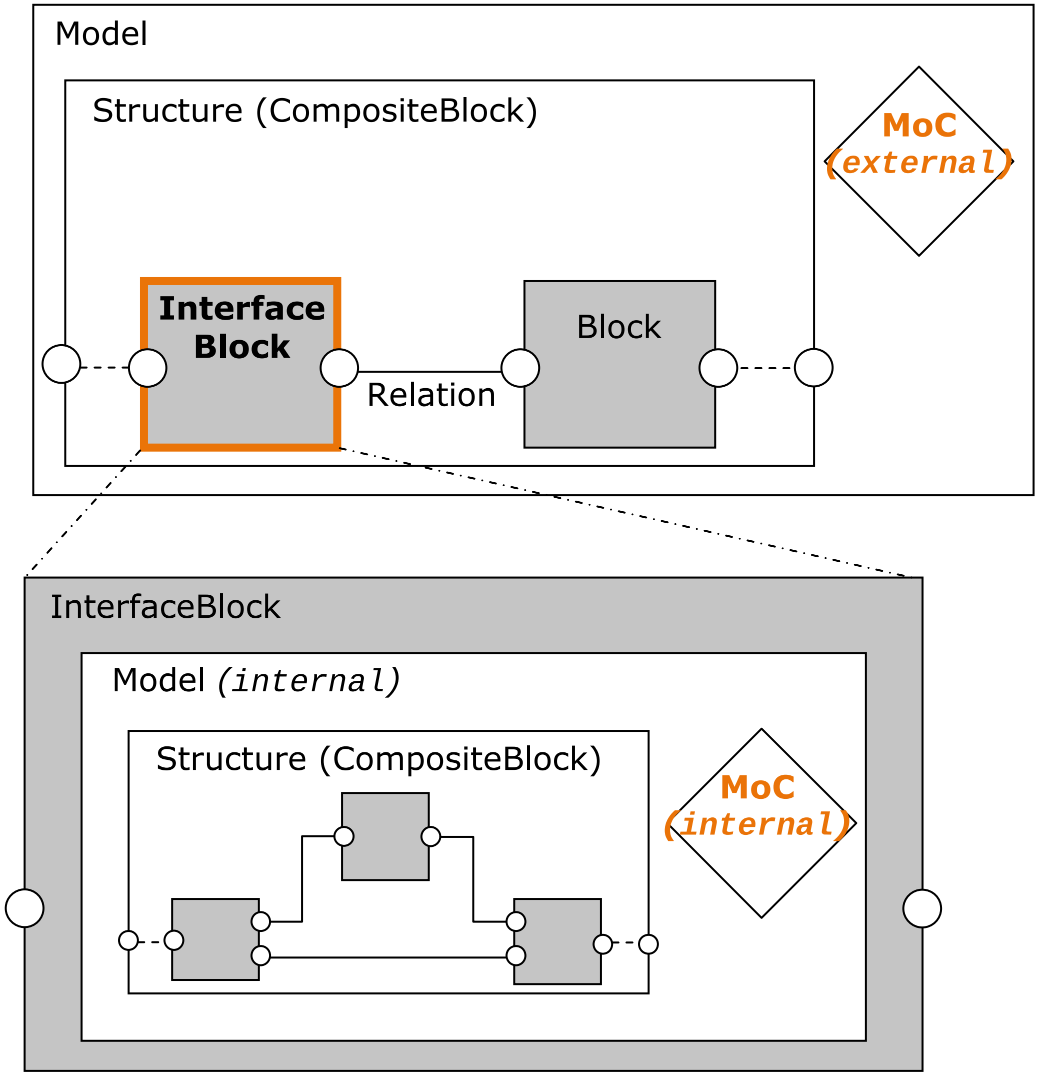
Hierarchy & glue

- Where is heterogeneity?



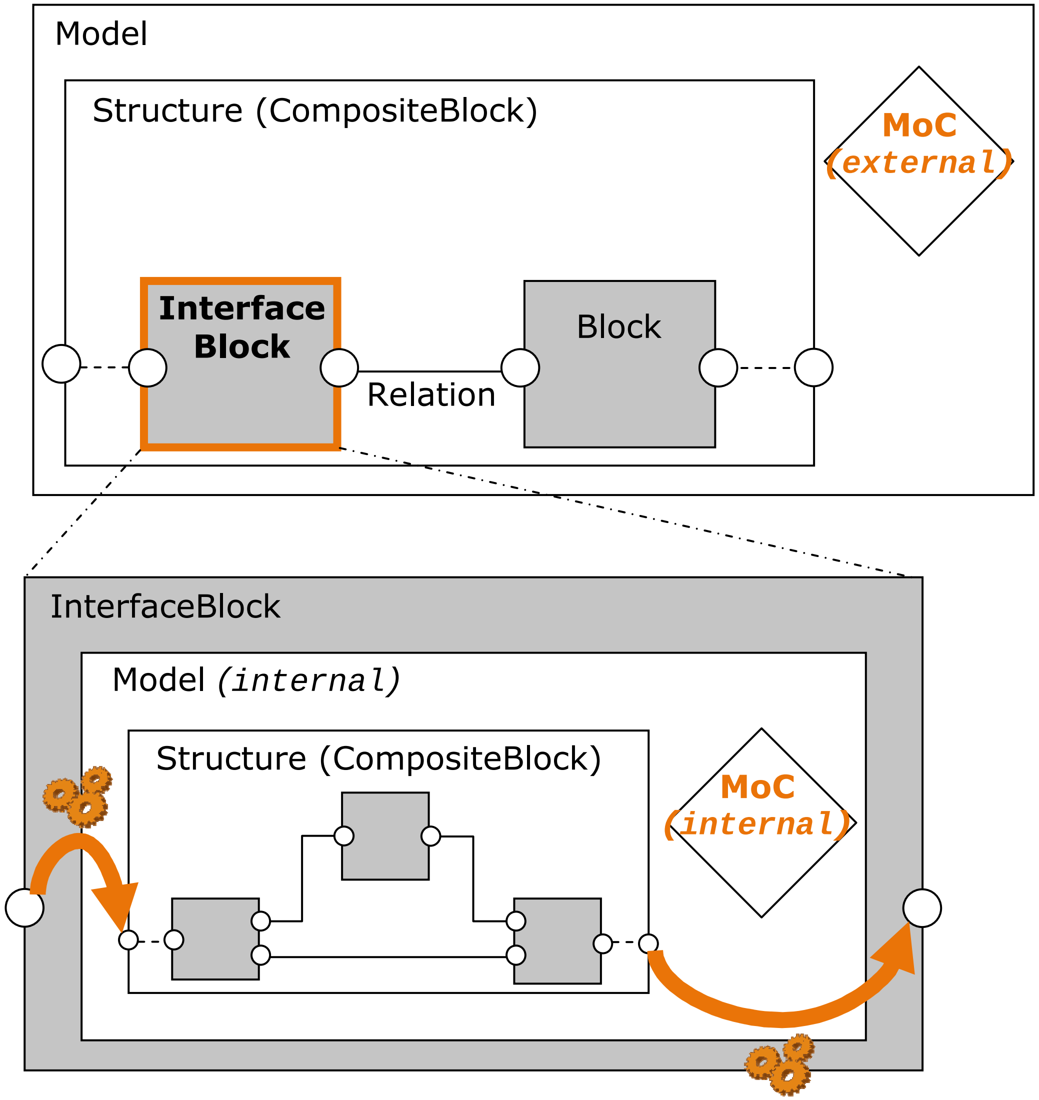
Hierarchy & glue

- Where is heterogeneity?



Hierarchy & glue

- Where is heterogeneity?



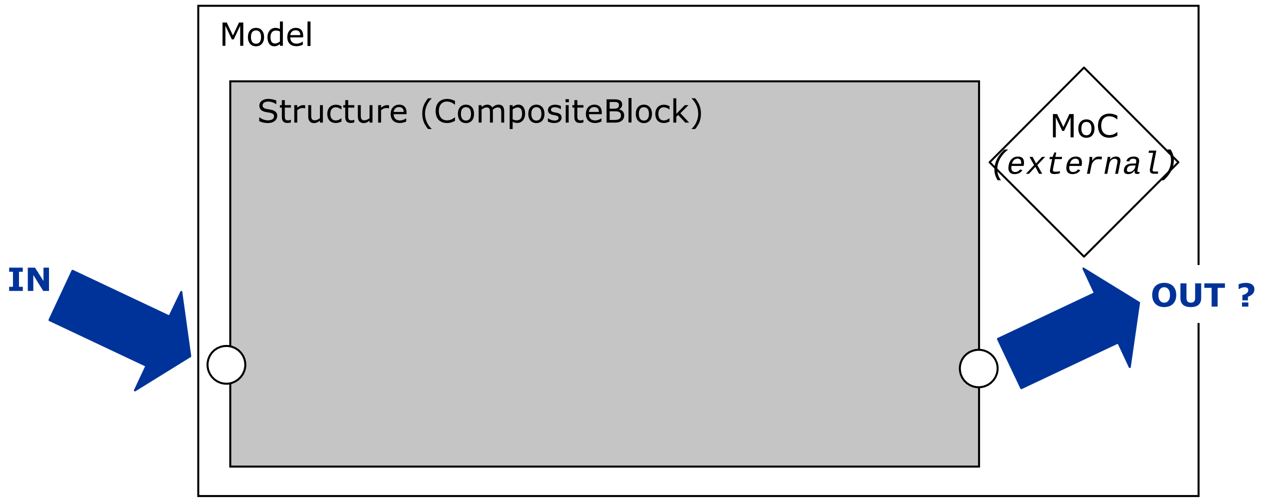
= Glue

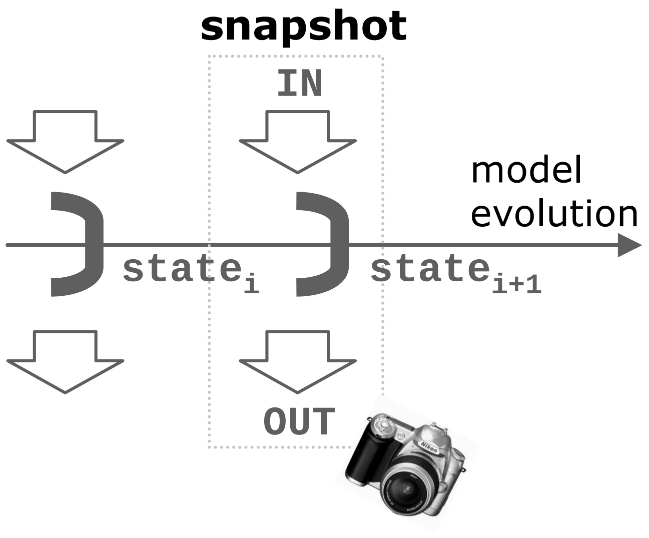
Agenda

1. Context, existing approaches & motivations
2. ModHel'X: underlying concepts
   - Abstract syntax, MoC, hierarchy & glue
   - Model execution
3. The coffee machine example
4. Discussion & conclusion

Model execution (observation)

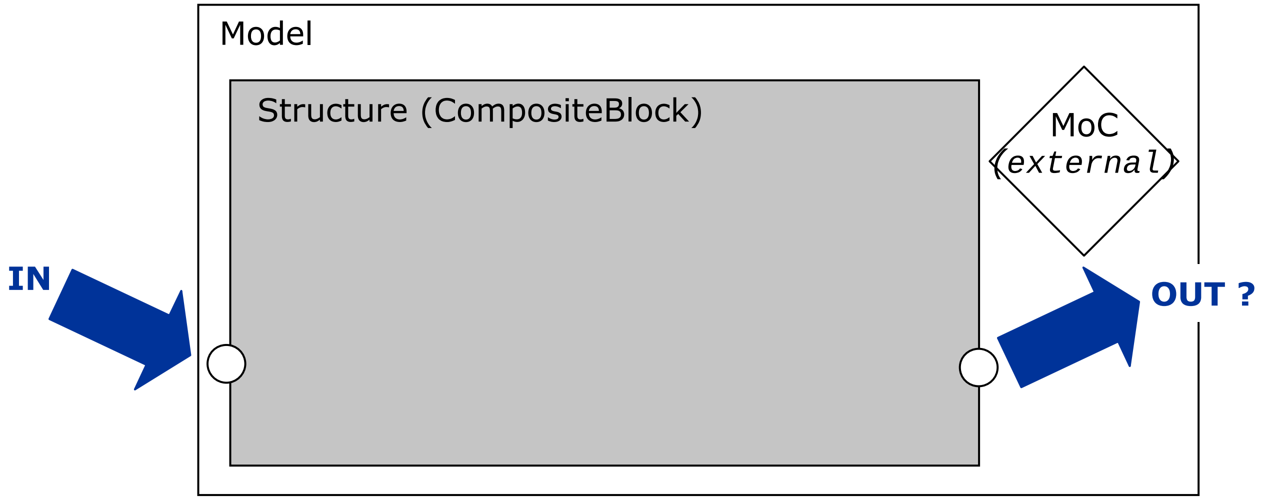
- Model execution = sequence of snapshots

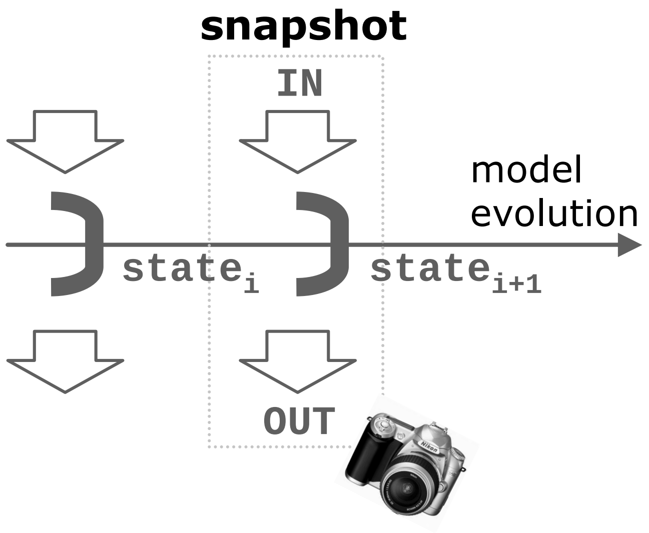




Model execution (observation)

- Model execution = sequence of snapshots



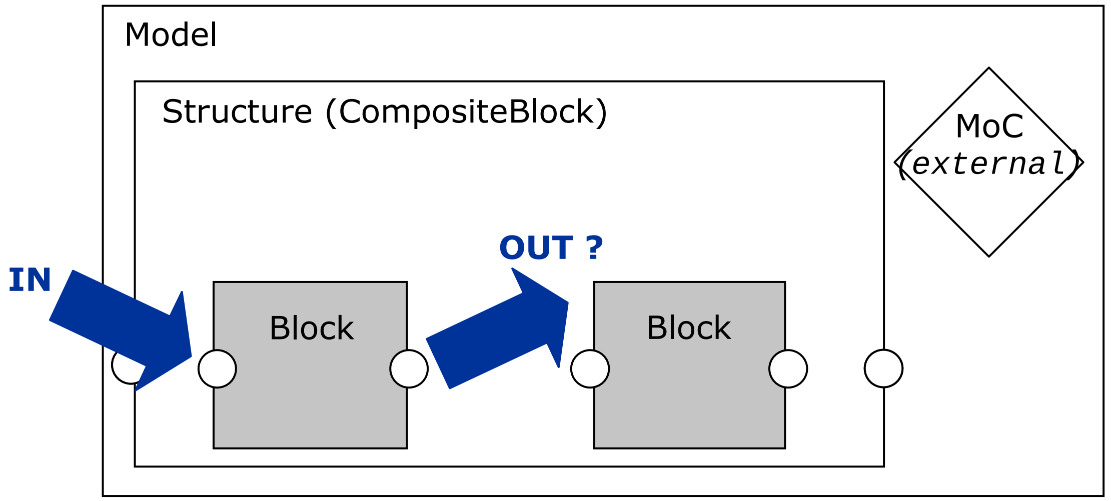


- When?
  - Regularly
  - When the time changes
  - When the environment changes
  - When the model changes (internally)

Depending on the MoCs involved!
(use of constraints)

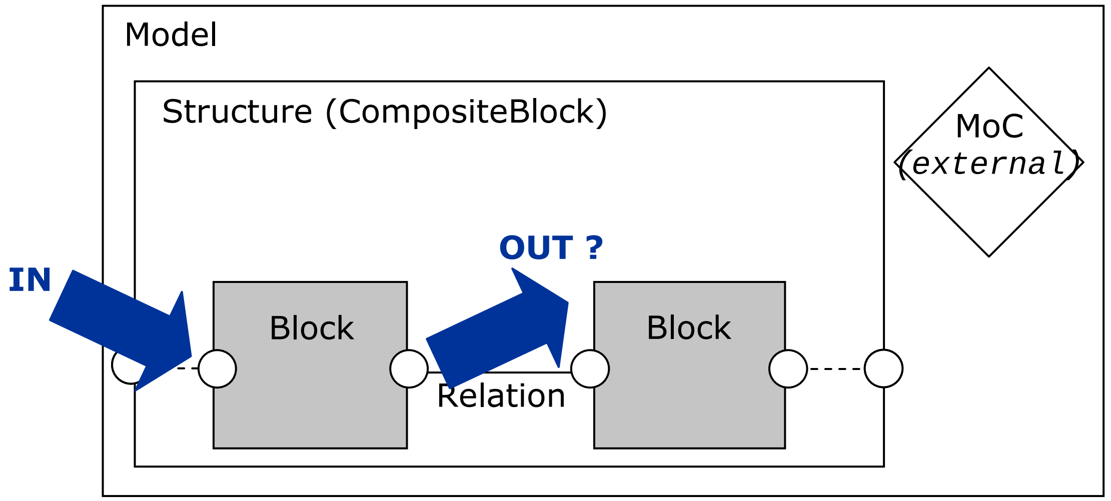
Model execution (observation)

- Snapshot = combination of block updates (observations)



Model execution (observation)

- Snapshot = combination of block updates (observations)

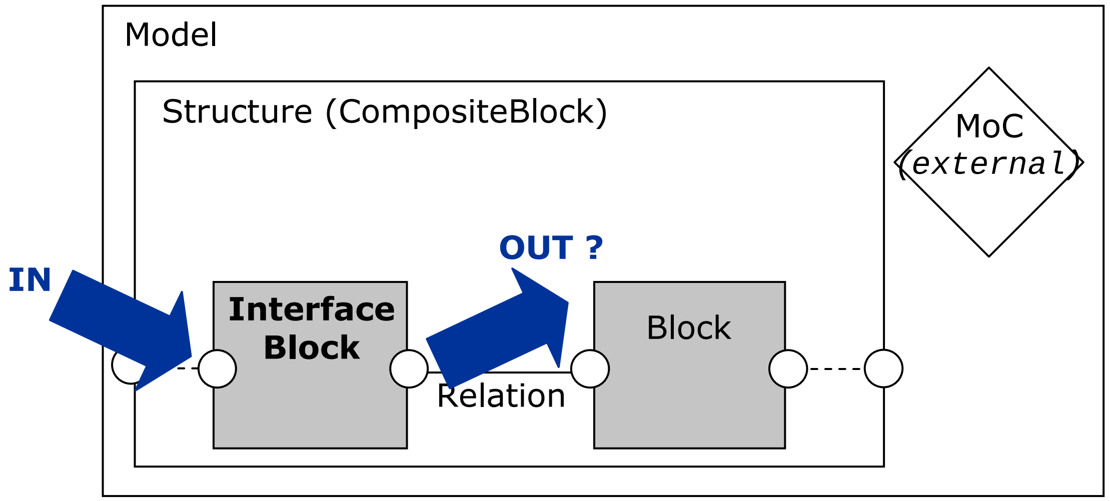


- In which order to update the blocks? (control and concurrency)
  - Topological order (e.g. DE)
  - Transitions (e.g. FSM), …
- How to propagate the results of the updates? (communication)
  - Timed events (e.g. DE)
  - Signal flows (e.g. SDF), …

Rules expressed by the MoC
(scheduling and propagation operations)

Model execution (observation)

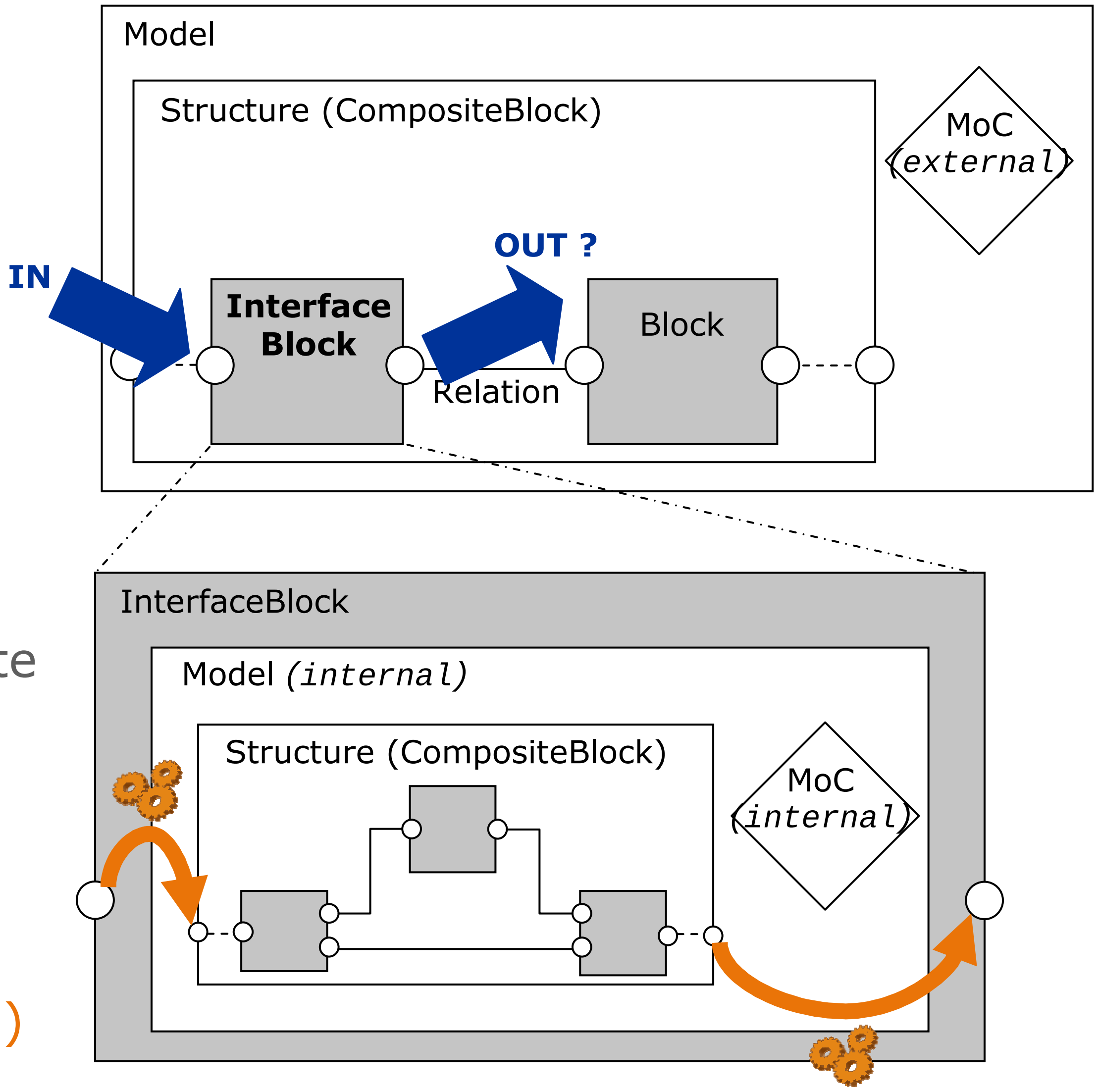
- Heterogeneity = hierarchy



ÉCOLE SUPÉRIEURE D'ÉLECTRICITÉ Supélec

Model execution (observation)

- Heterogeneity = hierarchy



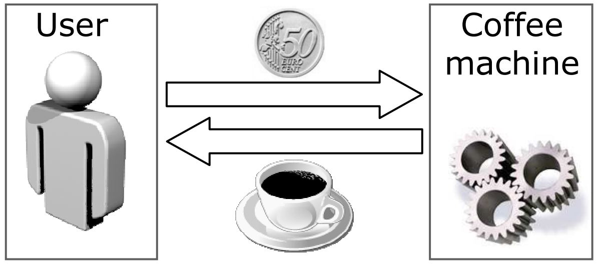
- Delegation of the update
- Adaptation
  - Data
  - Control
  - Time

Glue ( )

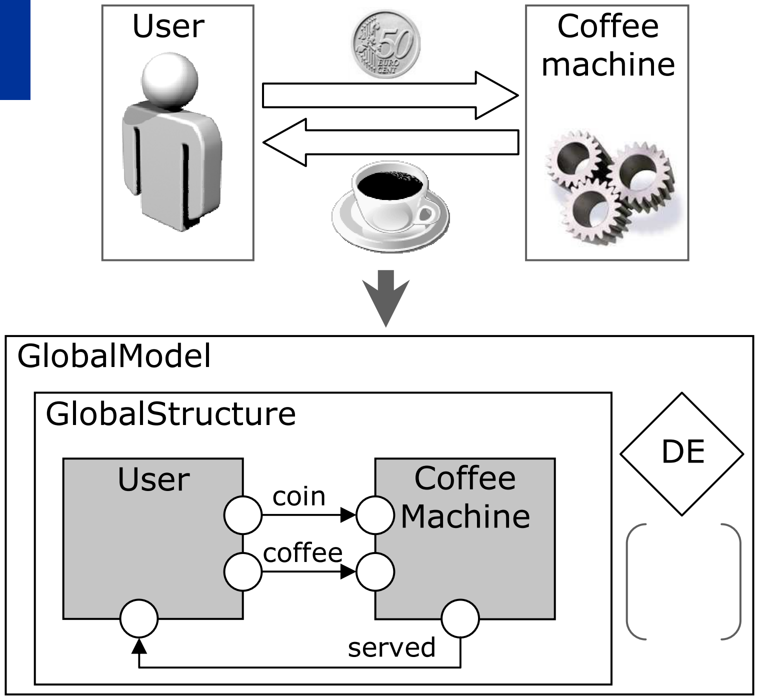
Agenda

1. Context, existing approaches & motivations
2. ModHel'X: underlying concepts
3. ▶ The coffee machine example
4. Discussion & conclusion

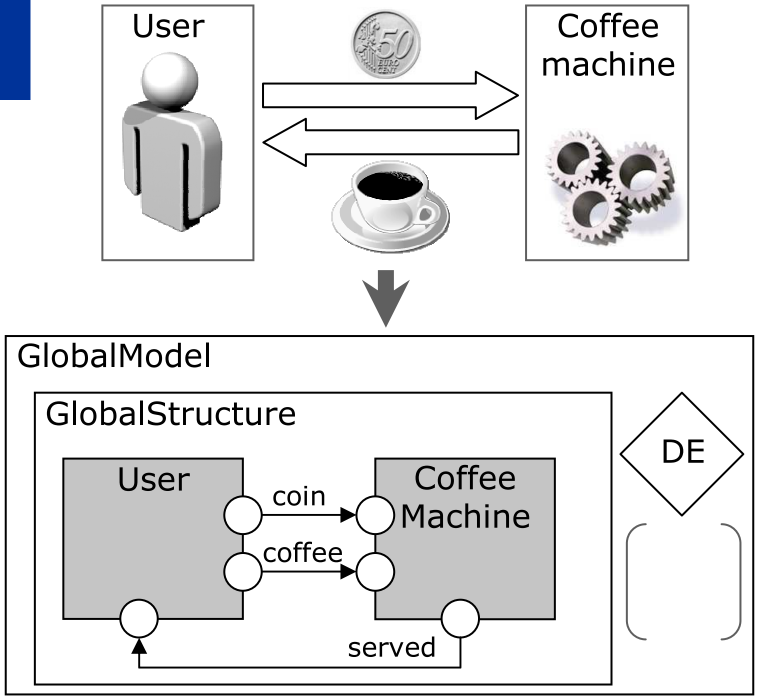
The coffee machine example



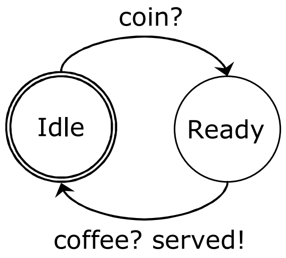
The coffee machine example



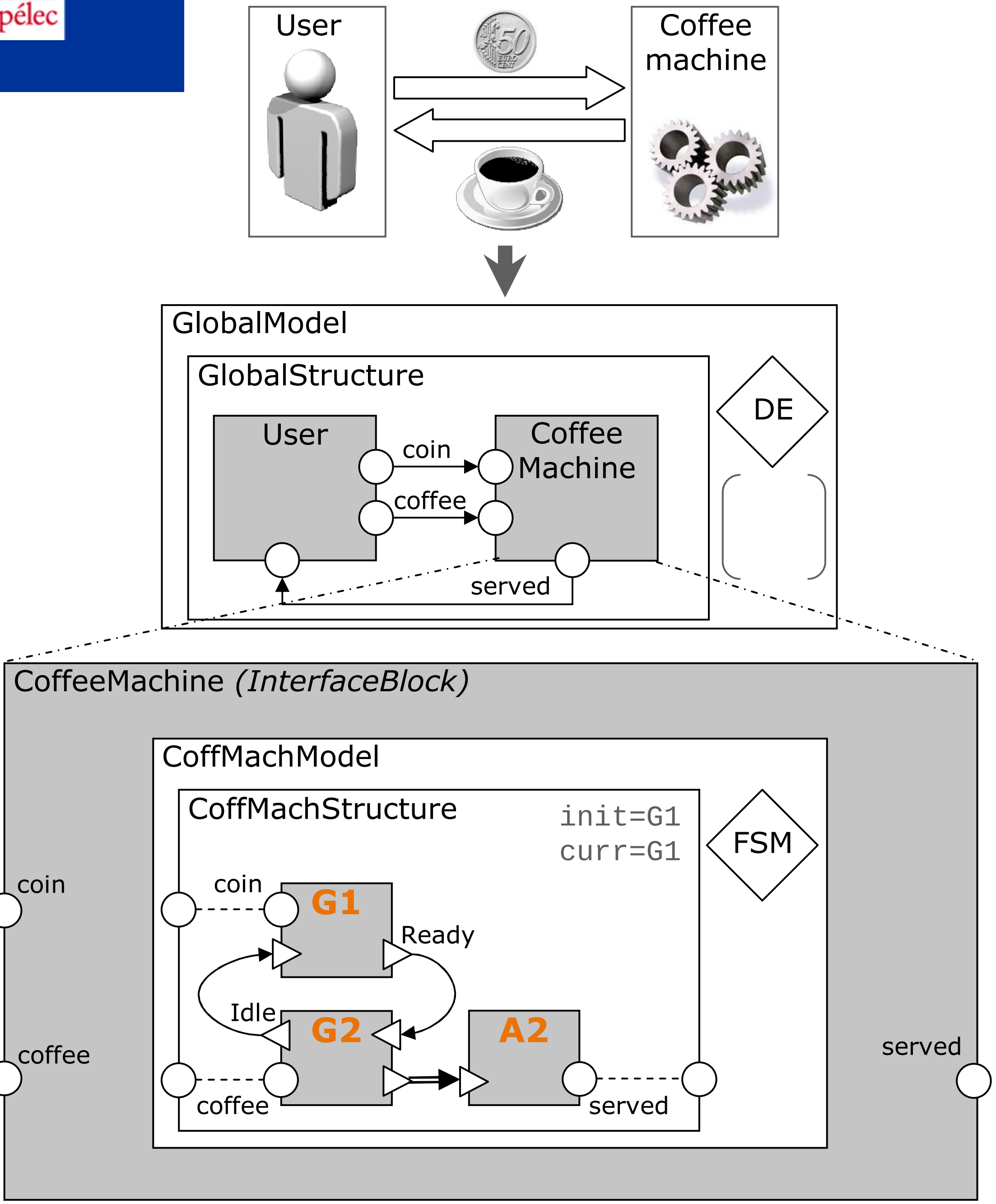
The coffee machine example



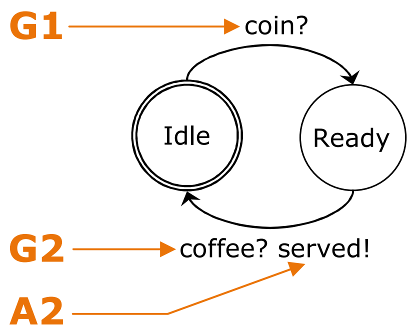
- Coffee machine automaton



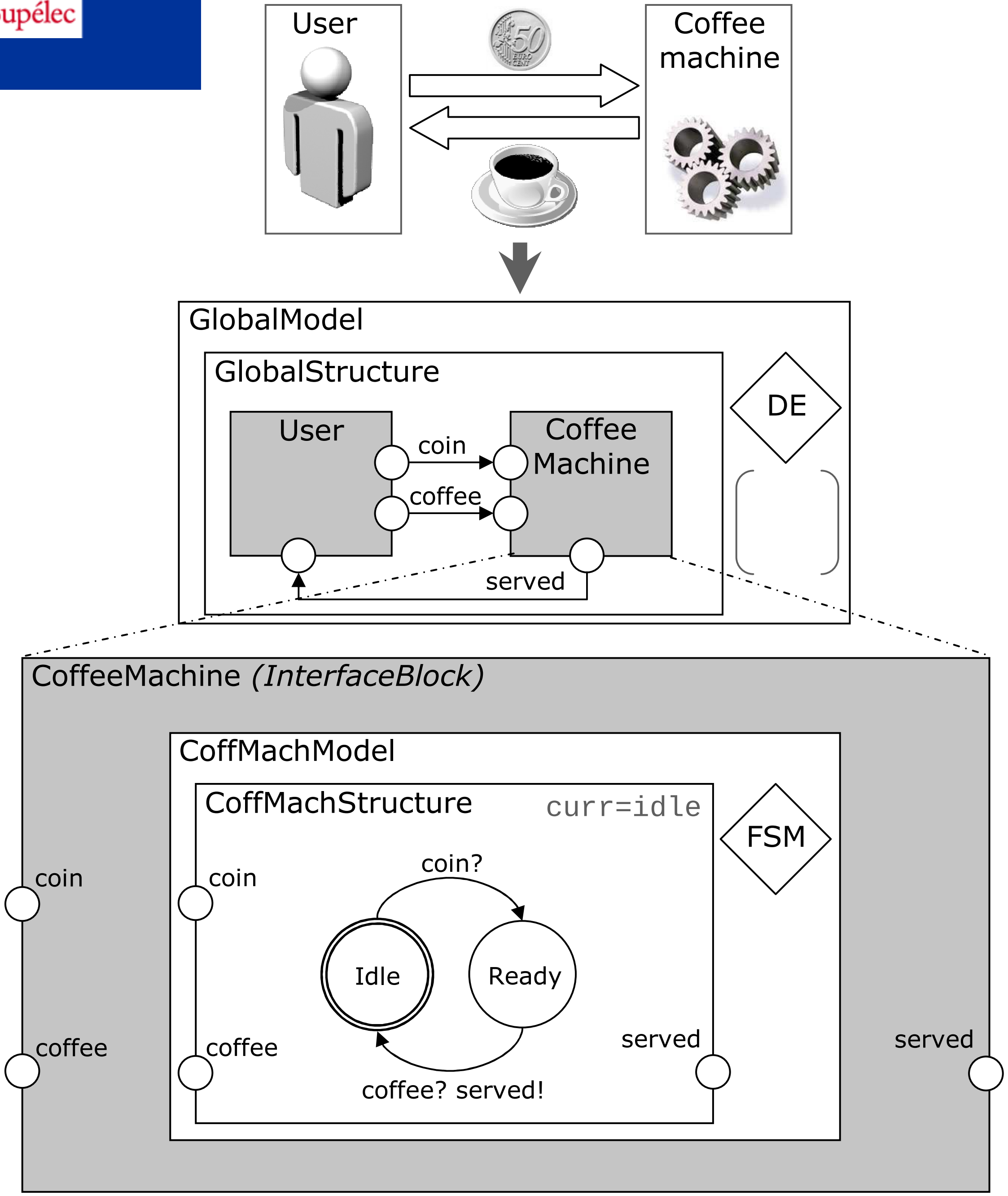
The coffee machine example



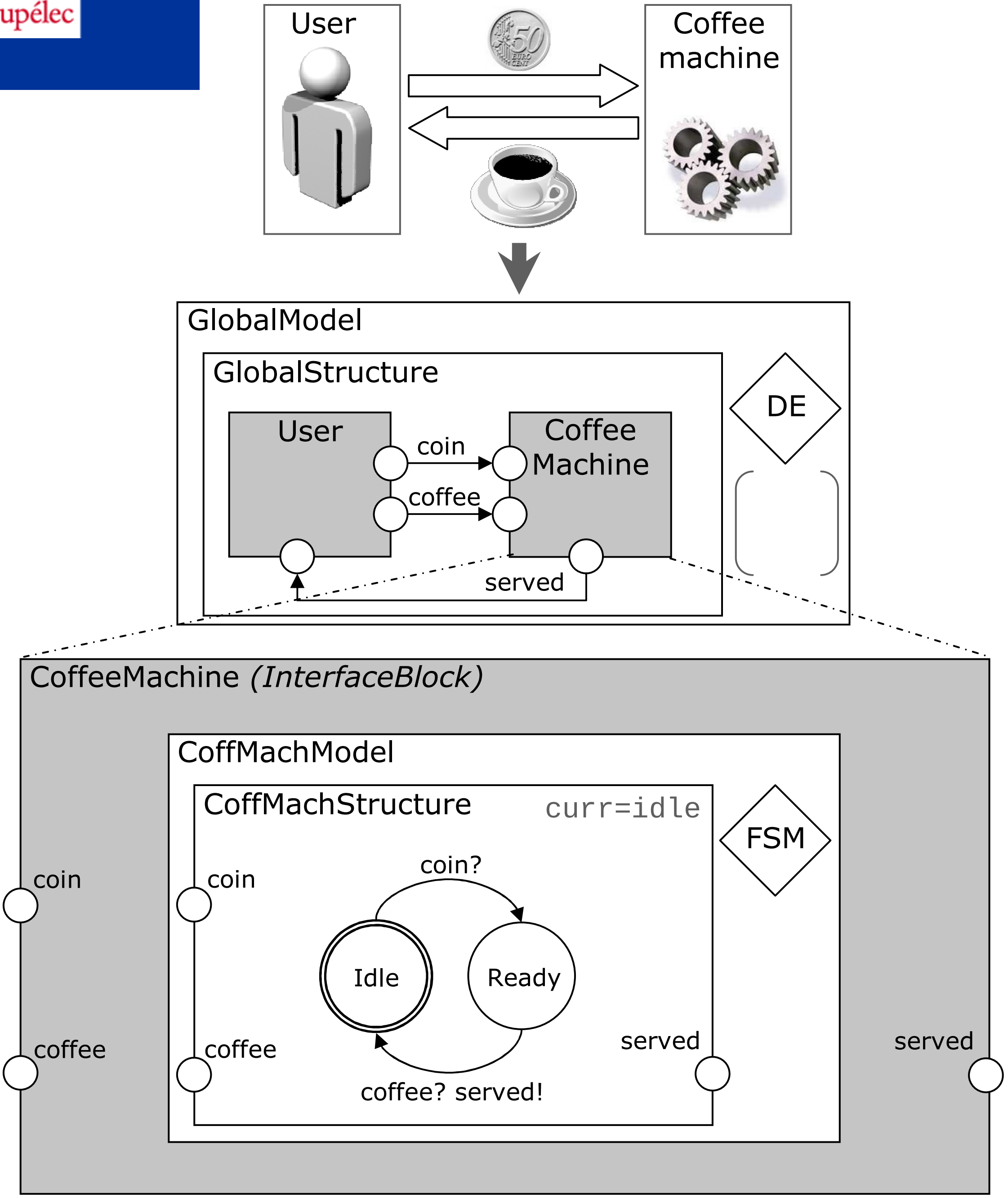
- Coffee machine automaton



The coffee machine example

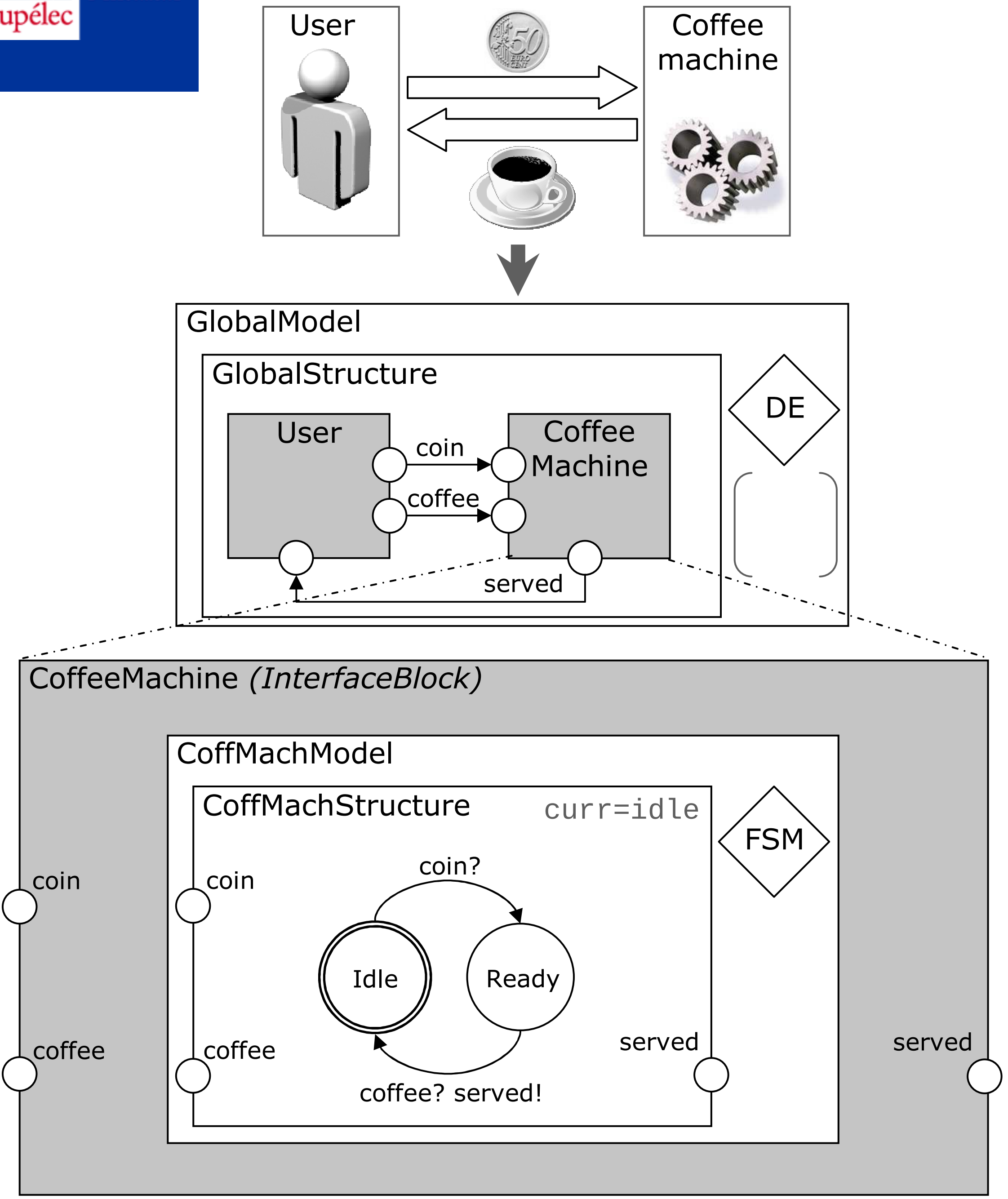


The coffee machine example



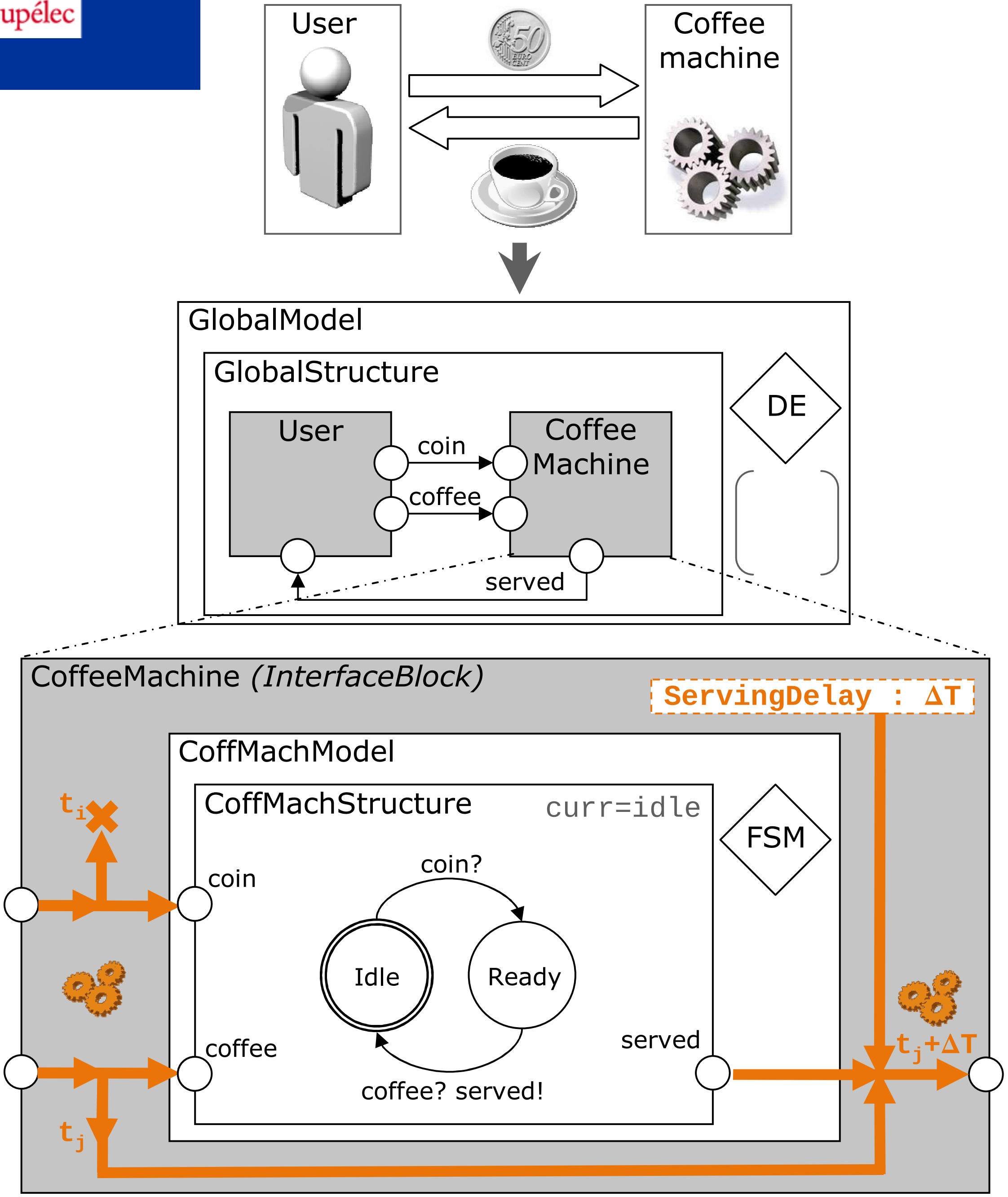
- Semantic adaptation: time “gluing”?

The coffee machine example



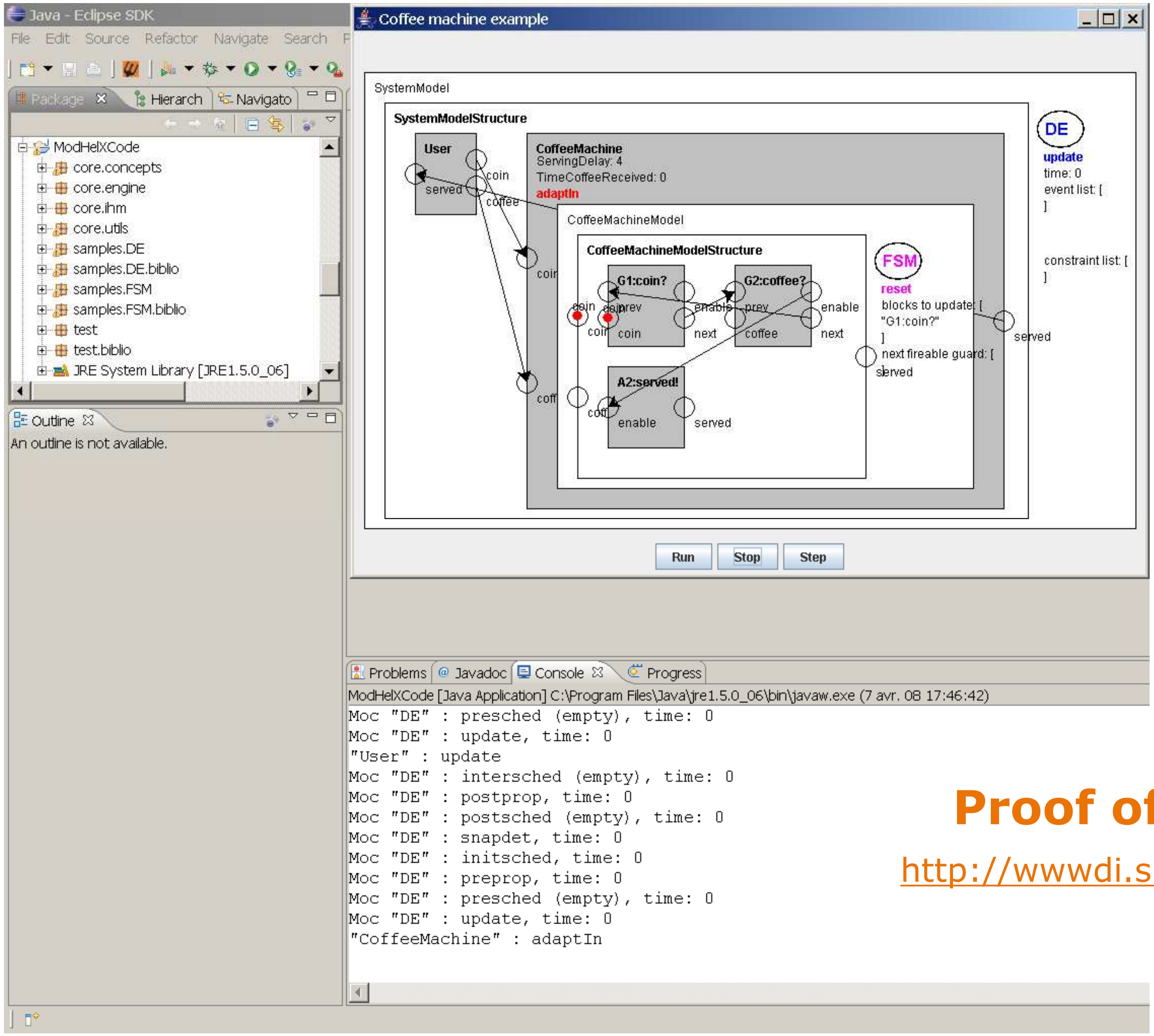
- Semantic adaptation: time “gluing”?
  - in: remove timestamps
  - out: add timestamps
  - ➡ which ones?

The coffee machine example



- Semantic adaptation: time "gluing"?
  - in: remove timestamps
  - out: add timestamps
    - which ones?

The coffee machine example



```
ModHelXCode [Java Application] C:\Program Files\Java\jre1.5.0_06\bin\javaw.exe (7 avr. 08 17:46:42)
Moc "DE" : presched (empty), time: 0
Moc "DE" : update, time: 0
"User" : update
Moc "DE" : intersched (empty), time: 0
Moc "DE" : postprop, time: 0
Moc "DE" : postsched (empty), time: 0
Moc "DE" : snapdet, time: 0
Moc "DE" : initsched, time: 0
Moc "DE" : preprop, time: 0
Moc "DE" : presched (empty), time: 0
Moc "DE" : update, time: 0
"CoffeeMachine" : adaptIn
```

Proof of concept demo

http://wwwdi.supelec.fr/logiciels/modhelx/

Agenda

1. Context, existing approaches & motivations
2. ModHel'X: underlying concepts
3. The coffee machine example
▶ 4. Discussion & conclusion

Discussion

- Use for tests:
  Simulation & real-time execution of heterogeneous models
  - Rapid prototyping
  - Execution of test scenarios
  - Generation of traces for analysis, …

- Supported MoCs
  - Continuous behaviors: numerical solving (approximation)
  - Cyclic dependencies: fixed point semantics (monotonicity…)
  - Non-determinism: "controlled" non-determinism (pseudo-random functions)

- How to add support for an additional language in ModHel'X?

once!

1. An expert of the modeling language describes:
   - The MoC corresponding to the language (structure + semantics)
   - Transformations from the original meta-model of the language to the ModHel'X meta-model
2. Experts define usual interaction patterns ("glues") for pairs of MoCs

Conclusion

- ModHel'X = an approach to multi-formalism modeling with
  - A generic meta-model for representing heterogeneous models
    - A specific structure for the explicit and flexible specification of the interactions between MoCs
  - A generic algorithm for executing heterogeneous models
    - A fixed frame for expressing MoCs
- Work in progress
  - Prototype based on the Eclipse Modeling Framework (EMF)
    - Several implemented MoCs
    - Working on the Synchronous DataFlow and UML StateCharts MoCs
  - Concrete syntax of our language (OMG ImperativeOCL – QVT)
    - Verbosity
    - Formal semantics
- Perspectives
  - Model based expression of glues
  - Combination of formal properties

References

[Kermeta] Muller, P.-A., F. Fleurey and J.-M. J´ez´equel, Weaving executability into object-oriented meta-languages, in: Proceedings of the 8th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS/UML 2005), 2005, pp. 264–278.

[ATOM^3] de Lara, J. and H. Vangheluwe, ATOM3: A tool for multi-formalism modelling and meta-modelling, in: 5th Fundamental Approaches to Software Engineering International Conference (FASE 2002), 2002, pp. 595–603.

[PtolemyII] Eker, J., J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs and Y. Xiong, Taming heterogeneity – the Ptolemy approach, Proceedings of the IEEE, Special Issue on Modeling and Design of Embedded Software 91 (2003), pp. 127–144.